

An Efficient Multiple-Groupcast Coded Multicasting Scheme for Finite Fractional Caching

Mingyue Ji*, Karthikeyan Shanmugam[¶], Giuseppe Vettigli[‡], Jaime Llorca[†], Antonia M. Tulino^{†‡}, Giuseppe Caire*[§]

[†]Alcatel Lucent, Bell labs, NJ. Email: {jaime.llorca, a.tulino}@alcatel-lucent.com

*EE Department, University of Southern California, CA. Email: mingyuej@usc.edu

[¶]ECE Department, University of Texas, TX. Email: karthiksh@utexas.edu

[‡]EE Department, Università di Napoli Federico II, Italy.

[§]EECS Department, Technische Universität Berlin, Germany. Email: caire@tu-berlin.de

Abstract—Coded multicasting has been shown to improve the caching performance of content delivery networks with multiple caches downstream of a common multicast link. However, the schemes that have been shown to achieve order-optimal performance require content items to be partitioned into a number of packets that grows exponentially with the number of users [1]. In this paper, we first extend the analysis of the achievable scheme in [2] to the case of heterogeneous cache sizes and demand distributions, providing an achievable scheme and an upper bound on the limiting average performance when the number of packets goes to infinity while the remaining system parameters are kept constant. We then show how the scheme achieving this upper bound can very quickly lose its multiplicative caching gain for finite content packetization. To overcome this limitation, we design a novel polynomial-time algorithm based on greedy local graph-coloring that, while keeping the same content packetization, recovers a significant part of the multiplicative caching gain. Our results show that the achievable schemes proposed to date to quantify the limiting performance, must be properly designed for practical finite system parameters.

I. INTRODUCTION

Recent studies [2]–[9] have been able to characterize the information theoretic limiting performance of several caching networks of practical relevance, in which network load scales inversely linear with cache size, showing great promise to accommodate the exponential traffic growth experienced in today’s communication networks [10].

We consider a network with n users, each with a cache of size M files, sharing a multicast link from a content source with access to a library of m files. For the worst-case demand setting, in which each user places a distinct file request, the authors in [7] presented a deterministic caching and coded multicasting scheme achieving the order-optimal transmission rate $^1 \Theta(\min\{\frac{m}{M}, m, n\})$.² However, the scheme in [7] requires a centralized caching policy and each file to be partitioned into a number of packets that grows exponentially with the number of users. In [8], the authors presented an alternative scheme for the same network that

uses a simpler decentralized random caching policy while a more complex coded multicasting scheme requiring a number of computations that grows exponentially with the number of users. Nonetheless, to guarantee the same rate, the file size (or equivalently the number of packets per file) is required to go to infinity. In [2], the authors extended the analysis to the case in which each user places $L \in \{1, \dots, m\}$ simultaneous file requests and provided an order-optimal delivery scheme based on local graph coloring [11] that is able to optimally combine the gains from coded and naive multicasting depending on the level of overlapping created by the multiple per-user requests. In [5], the authors considered the case in which user demands are characterized by a popularity distribution, and proposed a scheme consisting of a random popularity-based (RAP) caching policy and a chromatic-number index coding (CIC) multicasting scheme, referred to as RAP-CIC, proved to be order-optimal in terms of average rate. In order to analytically quantify the performance of RAP-CIC, the authors in [5] resorted to a polynomial-time approximation of CIC, referred as greedy constrained coloring (GCC) that guarantees the order-optimal rate in the asymptotic regime of infinite packetization. Using RAP-GCC, the authors further provided the regions of the system parameters, characterized by low popularity skewness and large aggregate cache size $nM > m$, in which multiplicative caching gains are potentially achievable.

It is then of key importance to understand if using any of above mentioned schemes, the promising multiplicative caching gain can be preserved in practical settings with finite file packetization. In this paper, we try to address this question focusing on a non-homogenous caching network with a shared multicast link, where users make $L \in \{1, \dots, m\}$ requests according to possibly different demand distributions and have possibly different cache sizes. As shown in Fig. 1, this scenario can be motivated by the presence of both user caches and cache-enabled small cell base stations, each of which serving a set of users. In this case, each small cell base station can be modeled as a user cache placing multiple requests. To this end, we first introduce RAP-GCLC (RAP-greedy constrained local coloring) as an extension to RAP-GCC in order to quantify the average performance of the to above non-homogenous shared link network. Next, we focus on the

¹We define transmission rate in terms of number of file transmissions or number of file-unit capacity channel-uses.

²Given two functions f and g , we say that: 1) $f(n) = O(g(n))$ if there exists a constant c and integer N such that $f(n) \leq cg(n)$ for $n > N$ 2) $f(n) = \Theta(g(n))$ if $f(n) = O(g(n))$ and $g(n) = O(f(n))$.

regime of finite file packetization and numerically show that RAP-GCLC cannot guarantee the asymptotic order-optimal performance. Consequently, we introduce a novel algorithm referred to as RAP-HgLC (RAP-hierarchical greedy local coloring), which is shown to recover a significant part of the asymptotic multiplicative caching gain. This algorithm's running time is quadratic in the number of packets.

The paper is organized as follows. Section II introduces the network model and problem formulation. The achievable caching and coded delivery scheme, along with the general upper bound on the average achievable rate are presented in Section III. Section IV describes the proposed polynomial-time delivery scheme. Finally, Section V presents the simulation results and related discussions.

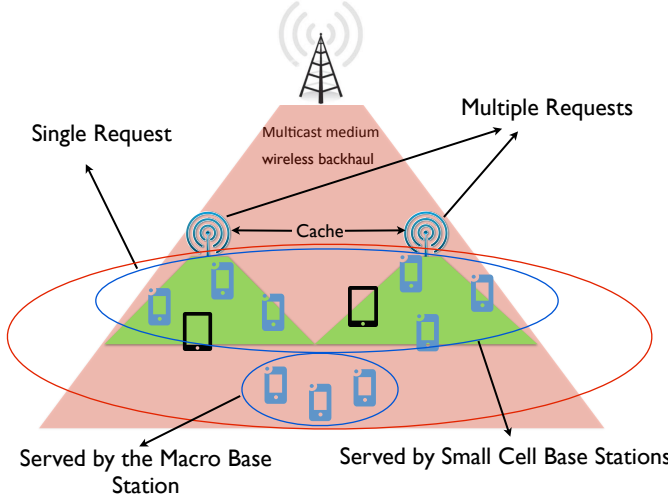


Fig. 1. An example of the network model, which consists of a source node (base station in this figure) having access to the content library and connected to the users via a shared bottleneck (multicast) link. Each user (end users and small cell base stations) may have different cache size and request a different number of files according to their own demand distribution.

II. NETWORK MODEL AND PROBLEM FORMULATION

We consider a network consisting of a source node with access to a content library $\mathcal{F} = \{1, \dots, m\}$ of files with size F bits, and n user nodes $\mathcal{U} = \{1, \dots, n\}$. We assume the source node communicates to the user nodes through a shared multicast link of finite capacity C . Without loss of generality, we can assume $C = F$ bits/unit time and measure the transmission rate of the scheme in units of time necessary to deliver the requested messages to the users. User $u \in \mathcal{U}$ has a storage capacity of size $M_u F$ bits (i.e., M_u files). The channel between the source and all the users follows a shared error-free deterministic model. User u makes L_u file requests, each of which follows a probability distribution $q_{f,u}$, where $q_{f,u} \in [0, 1]$ and $\sum_{f=1}^m q_{f,u} = 1$ (i.e., for each of the L_u requests of user u , file f is chosen with probability $q_{f,u}$). All the file requests (by one user or across users) are assumed to be placed independent of each other. We denote $\mathbf{Q} = [q_{f,u}]$, $u \in \{1, \dots, n\}$, $f \in \{1, \dots, m\}$, as the demand distribution. Let the requested files by user u be $\mathbf{f}_u = [f_{1,u}, f_{2,u}, \dots, f_{L_u,u}]$. The goal is to design a content distribution scheme (i.e., determine the information

stored in the user caches and the multicasted codeword to be sent to all users through the shared link) such that all demands are satisfied with probability 1 and the expected rate $R(\mathbf{Q})$ is minimized.³ The expectation is over the demand distribution \mathbf{Q} . We denote the minimum achievable expected rate by $R^*(\mathbf{Q})$.

III. ACHIEVABLE SCHEME

In this section, we present an achievable scheme based on random popularity-based caching and index coding based delivery.

A. Caching Placement

We partition each file into B equal-size packets, represented as symbols of $\mathbb{F}_{2^{F/B}}$, where F/B is sufficiently large (see later). Let \mathbf{C} and \mathbf{W} denote the realizations of the *packet level* caching and demand configurations, respectively, where $\mathbf{C}_{u,f}$ denotes the packets of file f cached at user u , and $\mathbf{W}_{u,f}$ denotes the packets of file f requested by user u . We let each user fill its cache independently (and therefore in a decentralized way) by knowing the caching distribution $\mathbf{P} = [p_{f,u}]$, $u \in \{1, \dots, n\}$, $f \in \{1, \dots, m\}$, with $\sum_{f=1}^m p_{f,u} = 1, \forall u$ and $0 \leq p_{f,u} \leq 1/M_u, \forall f$. The caching placement is shown in Algorithm 1.

Algorithm 1 Random Popularity-Based Caching (RAP)

- 1: **for all** $f \in \mathcal{F}$ **do**
- 2: Each user u caches a subset $(\mathbf{C}_{u,f})$ of $p_{f,u} M_u B$ distinct packets of file f uniformly at random.
- 3: **end for**
- 4: **return** $\mathbf{C} = [\mathbf{C}_{u,f}]$, $u \in \{1, \dots, n\}$, $f \in \{1, \dots, m\}$.

B. Coded Multicast Delivery

Our coded delivery scheme is based on local chromatic number index coding [2], [11]. The directed conflict graph $\mathcal{H}_{\mathbf{C}, \mathbf{W}} = (\mathcal{V}, \mathcal{E})$ is constructed as follows:

- Consider each packet requested by each user as a distinct vertex in $\mathcal{H}_{\mathbf{C}, \mathbf{W}} = (\mathcal{V}, \mathcal{E})$. Hence, each vertex $v \in \mathcal{V}$ is uniquely identified by the pair $\{\rho(v), \mu(v)\}$ where $\rho(v)$ indicates the packet identity associated to the vertex and $\mu(v)$ represents the user requesting it.
- For any pair of vertices v_1, v_2 , we say that vertex (packet) v_1 interferes with vertex v_2 if the packet associated to the vertex v_1 , $\rho(v_1)$, is not in the cache of the user associated to vertex v_2 , $\mu(v_2)$, and $\rho(v_1)$ and $\rho(v_2)$ do not represent the same packet. Then, draw a directed edge from vertex v_2 to vertex v_1 if v_1 interferes with v_2 .

We focus on encoding functions of the following form: for the request vectors \mathbf{f}_u , $u \in \mathcal{U}$, the multicast codeword is given by

$$X_{\{\mathbf{f}_u, u \in \mathcal{U}\}} = \sum_{v \in \mathcal{V}} \omega_v \mathbf{g}_v = \mathbf{G}\boldsymbol{\omega}, \quad (1)$$

³The expected rate is defined as the average minimum number of file transmissions.

where ω_v is the binary vector corresponding to packet v , represented as a (scalar) symbol of the extension field $\mathbb{F}_{2^{F/B}}$, the ν -dimensional vector $\mathbf{g}_v \in \mathbb{F}_{2^{F/B}}^\nu$ is the coding vector of packet $\rho(v)$ and where we let $\mathbf{G} = [\mathbf{g}_1, \dots, \mathbf{g}_{|\mathcal{V}|}]$ and $\boldsymbol{\omega} = [\omega_1, \dots, \omega_{|\mathcal{V}|}]^T$. The number of columns ν of \mathbf{G} yields the number of packet transmissions. Hence, the transmission rate is given by ν/B file units. To find the desired ν , we introduce the definition of the local chromatic number:

Definition 1: (Local Chromatic Number) The directed local chromatic number of a directed graph \mathcal{H}^d is defined as:

$$\chi_{lc}(\mathcal{H}^d) = \min_{\mathbf{c} \in \mathcal{C}} \max_{v \in \mathcal{V}} |\mathcal{C}(\mathcal{N}^+(v))| \quad (2)$$

where \mathcal{C} denotes the set of all vertex-colorings of \mathcal{H} , with \mathcal{H} indicating the undirected version of \mathcal{H}^d ,⁴ \mathcal{V} denotes the vertices of \mathcal{H}^d , $\mathcal{N}^+(v)$ is the closed out-neighborhood of vertex v ,⁵ and $\mathcal{C}(\mathcal{N}^+(v))$ is the total number of colors in $\mathcal{N}^+(v)$ for the given coloring \mathbf{c} . \diamond

It can be shown that, for sufficiently large F/B , there exists a \mathbf{G} such that a valid index code can be found whose length is equal to the local chromatic number, and whose corresponding transmission rate is given by $\chi_{lc}(\mathcal{H}^d)/B$. We refer to this coding scheme as LCIC (local chromatic index coding). The design of \mathbf{G} is given by [2], [11].⁶ and an example is given in the following:

Example 1: We consider a network with $n = 3$ users denoted as $\mathcal{U} = \{1, 2, 3\}$ and $m = 3$ files denoted as $\mathcal{F} = \{A, B, C\}$. We assume $M = 1$ and sub-packetize each file into three packets. For example, $A = \{A_1, A_2, A_3\}$. Let $p_{A,u} = \frac{1}{3}$, $p_{B,u} = \frac{1}{3}$ and $p_{C,u} = \frac{1}{3}$ for $u \in \{1, 2, 3\}$, which means that one packet from each of A, B, C will be stored in each user's cache. We assume a caching realization \mathbf{C} is given by: user u caches $\{A_u, B_u, C_u\}$ ($\mathbf{C}_{u,A} = \{A_u\}$, $\mathbf{C}_{u,B} = \{B_u\}$, $\mathbf{C}_{u,C} = \{C_u\}$). We let each user make one request. Specifically, we let user 1 request A , user 2 request A and user 3 request B ($\mathbf{f}_1 = \{A\}$, $\mathbf{f}_2 = \{A\}$, $\mathbf{f}_3 = \{B\}$) such that $\mathbf{W}_{1,A} = \{A_2, A_3\}$, $\mathbf{W}_{2,A} = \{A_1, A_3\}$, $\mathbf{W}_{3,B} = \{B_1, B_2\}$. The conflict graph and the corresponding coloring are shown in Fig. 2. We can see that the total number of colors needed, the chromatic number in this case, is 5, while the local coloring number, or the local chromatic number in this case, is 4. We construct \mathbf{G} by using the generator matrix of a $(5, 4)$ MDS code, which is given by:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}. \quad (3)$$

Then, we allocate the same vector to the vertex (packet) with the same color as shown in Fig. 2. Hence, the transmitted

codeword is given by $A_1 \oplus A_2$, $A_1 \oplus A_3$, $A_1 \oplus B_1$, $A_1 \oplus B_2$, of length $4/3$ file units. \diamond

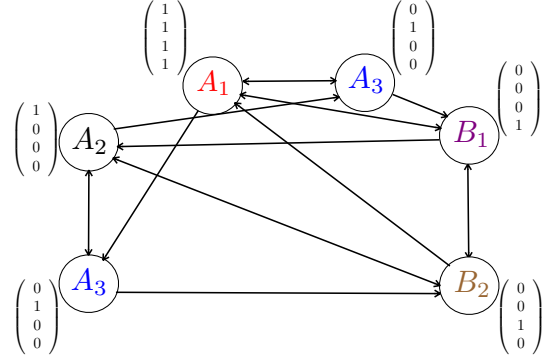


Fig. 2. An illustration of the directed conflict graph and the corresponding index code. The coloring of the graph is given by the colors of the fonts. The total number of colors is 5, and the local coloring number is 4.

It is worth noticing that given \mathbf{C} and \mathbf{W} , for any valid coloring scheme \mathbf{c} of $\mathcal{H}_{\mathbf{C}, \mathbf{W}}$, and its associated local coloring number, for sufficiently large F/B , there always exists an index code \mathbf{G} , such that the total number of transmissions in terms of packets is given by the local coloring number $\max_{v \in \mathcal{V}} |\mathcal{C}(\mathcal{N}^+(v))|$, and the corresponding transmission rate is $\max_{v \in \mathcal{V}} |\mathcal{C}(\mathcal{N}^+(v))|/B$.

C. Achievable Expected Rate

Given n, m, M and the demand distribution \mathbf{Q} , our goal is to find the caching distribution \mathbf{P} that minimizes the expected rate $R^{\text{LCIC}}(\mathbf{P}, \mathbf{Q}) \triangleq \lim_{B \rightarrow \infty} \mathbb{E}[\chi_{lc}(\mathcal{H}_{\mathbf{C}, \mathbf{W}})/B]$.⁷ Let $L = \max_u L_u$ and order $L_u, u \in \mathcal{U}$ as a decreasing sequence $L_{[1]} \geq L_{[2]} \geq L_{[3]} \geq \dots, L_{[n]}$, where $L_{[i]}$ is the i -th largest L_u and $[i] = u$ for some $u \in \mathcal{U}$. It can be seen that $L_{[1]} = \max_u L_u$ and $L_{[n]} = \min_u L_u$. Let $n_j = \sum_{[i]} 1\{L_{[i]} - j \geq 0\}$, where $1 \leq j \leq L_{[1]}$ and $1\{\cdot\}$ is the indicator function. Let $\mathcal{U}_{n_j} = \{[i] \in \mathcal{U} : 1\{L_{[i]} - j \geq 0\}\}$. In the next theorem we provide an upper bound of $R^{\text{LCIC}}(\mathbf{P}, \mathbf{Q})$ given by the rate achievable with a greedy constrained local coloring (GCLC) scheme which is described in details in the next section (see Section IV-A) and can be seen as a generalization of the GCC scheme presented in [5].

Theorem 1: For any given m, n, M_u , and \mathbf{Q} , when $B \rightarrow \infty$, the expected rate $R^{\text{LCIC}}(\mathbf{P}, \mathbf{Q})$ achieved by a content distribution scheme that uses caching policy in Algorithm 1 with caching distribution $\{\mathbf{P} = [p_{f,u}] : \sum_{f=1}^m p_{f,u} = 1, \forall u; 0 \leq p_{f,u} \leq 1/M_u, \forall f, u\}$, and LCIC transmission, satisfies:⁸

$$R^{\text{LCIC}}(\mathbf{P}, \mathbf{Q}) \leq R^{\text{GCLC}}(\mathbf{P}, \mathbf{Q}) \triangleq \min\{\psi(\mathbf{P}, \mathbf{Q}), \bar{m} - \bar{M}\}, \quad (4)$$

⁴An edge is undirected if the edge is present in either direction.

⁵Closed out-neighborhood of vertex v includes vertex v and all the connected vertices via out-going edges of v .

⁶Instead of using local chromatic number it is also straightforward to use fractional local chromatic number to design the coding vector \mathbf{G} as illustrated in [2], [11].

⁷ $\mathcal{H}_{\mathbf{C}, \mathbf{W}}$ denotes the random conflict graph, which is a function of the random caching and demand configurations, \mathbf{C} and \mathbf{W} , respectively.

⁸A stronger version Theorem 1 states the following stronger result: $\lim_{F \rightarrow \infty} \mathbb{P}(\mathbb{E}[\chi_{lc}(\mathcal{H}_{\mathbf{C}, \mathbf{W}})/B] \leq R^{\text{GCLC}}(\mathbf{P}, \mathbf{Q}) + \epsilon) = 1$, where the expectation is taken over only the demand distribution \mathbf{Q} .

In (4),

$$\bar{m} = \sum_{f=1}^m \left(1 - \prod_{u=1}^n (1 - q_{f,u})^{L_u} \right), \quad (5)$$

and

$$\bar{M} = \sum_{f=1}^m \min_u p_{f,u} \left(1 - \prod_{u=1}^n (1 - q_{f,u})^{L_u} \right), \quad (6)$$

and

$$\begin{aligned} \psi(\mathbf{P}, \mathbf{Q}) &= \sum_{j=1}^L \sum_{\ell=1}^n \sum_{\mathcal{U}^\ell \subset \mathcal{U}_{n_j}} \sum_{f=1}^m \sum_{u \in \mathcal{U}^\ell} \\ &\quad \rho_{f,u,\mathcal{U}^\ell} (1 - p_{f,u} M_u)^{n_j - \ell + 1} (p_{f,u} M_u)^{\ell - 1}, \end{aligned}$$

where \mathcal{U}^ℓ denotes a set of users with cardinality ℓ and

$$\begin{aligned} \rho_{f,u,\mathcal{U}^\ell} &\triangleq \\ \mathbb{P}(f = \arg \max_{f_u \in \mathbf{f}(\mathcal{U}^\ell)} (p_{f,u} M_u)^{\ell - 1} (1 - p_{f,u} M_u)^{n_j - \ell + 1}), \end{aligned}$$

denotes the probability that f is the file whose $p_{f,u}$ maximizes the term $(p_{f,u} M_u)^{\ell - 1} (1 - p_{f,u} M_u)^{n_j - \ell + 1}$ among $\mathbf{f}(\mathcal{U}^\ell)$ (the set of files requested by \mathcal{U}^ℓ). \square

Under homogeneous demand distribution, cache size and number of request per user, we have the following corollary:

Corollary 1: Let $q_{f,u} = q_f$, $M_u = M$, $L_u = L$, $\forall u \in \mathcal{U}$ and $L = \{1, \dots, m\}$, then $p_{f,u} = p_f$, $\forall u \in \mathcal{U}$ and when $B \rightarrow \infty$, $R^{\text{LCIC}}(\mathbf{P}, \mathbf{Q})$ is given by (4), where

$$\bar{m} = \sum_{f=1}^m \left(1 - (1 - q_f)^{nL} \right), \quad (7)$$

and

$$\bar{M} = \sum_{f=1}^m p_f \left(1 - (1 - q_f)^{nL} \right), \quad (8)$$

and

$$\begin{aligned} \psi(\mathbf{P}, \mathbf{Q}) &= \\ L \sum_{\ell=1}^n \binom{n}{\ell} \sum_{f=1}^m \rho_{f,\ell} (1 - p_f M)^{n - \ell + 1} (p_f M)^{\ell - 1}, \end{aligned} \quad (9)$$

where $\rho_{f,\ell} \triangleq \mathbb{P}(f = \arg \max_{j \in \mathcal{D}} (p_j M)^{\ell - 1} (1 - p_j M)^{n - \ell + 1})$ denotes the probability that file f is the file whose p_f maximizes the term $((p_j M)^{\ell - 1} (1 - p_j M)^{n - \ell + 1})$ among \mathcal{D} , which is a set of ℓ i.i.d. demands distributed as \mathbf{q} . It can be seen that $\rho_{f,\ell}$ is easy to evaluate. \square

Due to space limitations, the proof of Theorem 1 is not included in this paper. Corollary 1 can be obtained directly from Theorem 1.

Using the explicit expression for $R^{\text{GCLC}}(\mathbf{P}, \mathbf{Q})$ in Theorem 1, we can optimize the caching distribution for a wide class of heterogeneous network models in order to minimize the number of transmissions. We use \mathbf{P}^* to denote the caching distribution that minimizes $R^{\text{GCLC}}(\mathbf{P}, \mathbf{Q})$. It is worth noticing that for the homogeneous case described above, where $q_{f,u} = q_f$, $M_u = M$, $L_u = 1$, $\forall u \in \mathcal{U}$, $R^{\text{GCLC}}(\mathbf{P}^*, \mathbf{Q})$ is indeed order optimal, as proved in [5].

IV. POLYNOMIAL-TIME ALGORITHMS

In this section, we propose two efficient coloring algorithms for coded multicasting in heterogeneous shared link caching networks. We first introduce a polynomial-time greedy constrained local coloring (GCLC) algorithm, which generalizes the greedy constrained coloring (GCC) used in [5] to quantify the order-optimal performance of homogeneous shared link networks in the asymptotic regime of $B \rightarrow \infty$. In fact, GCLC is the scheme whose asymptotic ($B \rightarrow \infty$) average rate for heterogeneous shared link networks has been used in (4) to upper bound the asymptotic rate of LCIC. It is also easy to verify that GCLC achieves the same performance as the algorithm given in [2] for the worst-case demand setting in homogeneous shared link networks.

We then present a novel coded multicasting algorithm called hierarchical greedy local coloring (HgLC) that fully exploits the structure of the problem and also exhibits polynomial-time complexity. In Section V, we show that for finite file packetization, while GCLC loses the multiplicative caching gain, HgLC is able to approach the limiting performance and recover a significant part of the multiplicative caching gain.

A. GCLC (Greedy Constrained Local Coloring)

The GCLC algorithm works by computing two valid local colorings of the conflict graph $\mathcal{H}_{\mathbf{C}, \mathbf{W}}$, referred to as GCLC₁ and GCLC₂. GCLC then compares the rate achieved by the two coloring solutions and constructs the transmission code based on the coloring with minimum rate.

Let \mathbf{W}_u be the set of requested packets by user u , and \mathbf{C}_u the set of cached packets by user u . We define $\mathcal{T}_v = \{\mu(v)\} \cup \{u \in \mathcal{U} : \rho(v) \in \mathbf{C}_u\}$. Then, GCLC₁ is given by Algorithm 2. Observe that GCLC₁ computes a valid coloring of the conflict graph $\mathcal{H}_{\mathbf{C}, \mathbf{W}}$ and its associated local coloring number. Note that both the outer while-loop starting at line 3 and the inner for-loop starting at line 6 iterate at most $|\mathcal{V}|$ times, respectively, while all other operations inside the loops take constant time. Therefore, the complexity of GCLC₁ is $O(|\mathcal{V}|^2)$, which is polynomial in $|\mathcal{V}|$ (or n, B).

Algorithm 2 GCLC₁

```

1: Let  $\mathcal{C} = \emptyset$ ;
2: Let  $\mathbf{c} = \emptyset$ ;
3: while  $\mathcal{V} \neq \emptyset$  do
4:   Pick an arbitrary vertex  $v$  in  $\mathcal{V}$ ; Let  $\mathcal{I} = \{v\}$ ;
5:   Let  $\mathcal{V}' = \mathcal{V} \setminus \{v\}$ ;
6:   for all  $v' \in \mathcal{V}'$  with  $|\mathcal{T}_{v'}| = |\mathcal{T}_v|$  do
7:     if {There is no edge between  $v'$  and  $\mathcal{I}$ } then
8:        $\mathcal{I} = \mathcal{I} \cup v'$ ;
9:     end if
10:  end for
11:  Color all the vertices in  $\mathcal{I}$  by  $c \notin \mathcal{C}$ ;
12:  Let  $\mathbf{c}[\mathcal{I}] = c$ ;
13:   $\mathcal{V} = \mathcal{V} \setminus \mathcal{I}$ ;
14: end while
15: return  $\max_{v \in \mathcal{V}} |\mathbf{c}(\mathcal{N}^+(v))|$  and the corresponding  $\mathbf{c}(\mathcal{N}^+(v))$ 
    for each  $v$ ;

```

Algorithm 3 HgLC₁

```

1:  $\mathcal{C} = \emptyset$ ;
2:  $\mathbf{c} = \emptyset$ ;
3: choose  $a \in [0, 1]$ 
4: choose  $b \in [0, 1]$ 
5: for all  $i = n, n-1, \dots, 2, 1$  do
6:   for all  $v \in \mathcal{G}_i$  and  $|\mathcal{K}_v| = i$  do
7:      $\mathcal{I} = \{v\}$ ;
8:     for all  $v' \in \mathcal{G}_i \setminus \mathcal{I}$  with  $|\mathcal{K}_{v'}| = |\mathcal{K}_v|$  do
9:       if {There is no edge between  $v'$  and  $\mathcal{I}$ } then
10:         $\mathcal{I} = \mathcal{I} \cup v'$ ;
11:       end if
12:     end for
13:     if  $|\mathcal{I}| = i$  then
14:       Color all the vertices in  $\mathcal{I}$  by  $c \notin \mathcal{C}$ ;
15:        $\mathbf{c}[\mathcal{I}] = c$ ,  $\mathcal{C} = \mathcal{C} \cup c$ ;
16:        $\mathcal{G}_i = \mathcal{G}_i \setminus \mathcal{I}$ ;
17:     end if
18:   end for
19:   for all  $v \in \mathcal{G}_i$  with  $v$  randomly picked from  $\mathcal{W}_1 \subset \mathcal{G}_i$  do
20:      $\mathcal{I} = \{v\}$ ;
21:      $\mathcal{Q}_i = \mathcal{G}_i \setminus \mathcal{I}$ ;
22:     for all  $v' \in \mathcal{Q}_i$  with  $v'$  randomly picked from  $\mathcal{W}_2 \subset \mathcal{Q}_i$ .
23:     do
24:       if {There is no edge between  $v'$  and  $\mathcal{I}$ } then
25:         $\mathcal{I} = \mathcal{I} \cup v'$ ;
26:         $\mathcal{Q}_i = \mathcal{Q}_i \setminus \{v'\}$ ;
27:       else
28:         $\mathcal{Q}_i = \mathcal{Q}_i \setminus \{v'\}$ ;
29:       end if
30:     end for
31:     if  $|\mathcal{I}| \geq i$  then
32:       Color all the vertices in  $\mathcal{I}$  by  $c \notin \mathcal{C}$ ;
33:        $\mathbf{c}[\mathcal{I}] = c$ ,  $\mathcal{C} = \mathcal{C} \cup c$ ;
34:        $\mathcal{G}_i = \mathcal{G}_i \setminus \mathcal{I}$ ;
35:     else
36:        $\mathcal{G}_i = \mathcal{G}_i \setminus \{v\}$ ,  $\mathcal{G}_{i-1} = \mathcal{G}_{i-1} \cup \{v\}$ ;
37:     end if
38:   end for
39:  $\mathbf{c} = \text{LocalSearch}(\mathcal{H}_{\mathbf{C}, \mathbf{W}}, \mathbf{c}, \mathcal{C})$ ;
40: return  $\max_{v \in \mathcal{V}} |\mathbf{c}(\mathcal{N}^+(v))|$  and the corresponding  $\mathbf{c}(\mathcal{N}^+(v))$ 
    for each  $v$ ;

```

On the other hand, GCLC₂ computes the minimum coloring of $\mathcal{H}_{\mathbf{C}, \mathbf{W}}$ subject to the constraint that only the vertices representing the same packet are allowed to have the same color. In this case, the total number of colors is equal to the number of distinct requested packets, and the coloring can be found in $O(|\mathcal{V}|^2)$. Then, it remains to find $\max_{v \in \mathcal{V}} |\mathbf{c}(\mathcal{N}^+(v))|$. It can be seen that this scheme achieves the same rate as sending linear random combinations of all requested packets.

This scheme, GCLC, was shown in [2] to be order-optimal when $B \rightarrow \infty$ for the homogeneous shared link network with L requests per user, in the worst-case demand setting.

B. Hierarchical greedy Local Coloring (HgLC)

Similar to GCLC, HgLC also works by first computing two valid local colorings of the conflict graph $\mathcal{H}_{\mathbf{C}, \mathbf{W}}$, referred to as HgLC₁ and HgLC₂. Then, the transmission code is constructed based on the coloring with minimum rate. In this

case, HgLC₂ is the same as GCLC₂, while HgLC₁ is described by Algorithm 3.

We guide the reader through Algorithm 3 in the following. Let $\mathcal{K}_v = \{\forall u \in \mathcal{U} : \rho(v) \in \mathbf{W}_u \cup \mathbf{C}_u\}$ and $\mathcal{G}_i = \{v : |\mathcal{K}_v| = i\}$. We consider \mathcal{G}_i to represent the i th hierarchy. For HgLC₁, we start from hierarchy n . First, we color a subset of vertices in \mathcal{G}_n with the same color if: they have the same $|\mathcal{K}_v| = n$, the cardinality of such set is n , and there are no links between any two vertices of such set in the conflict graph. Then, we merge the uncolored vertices in \mathcal{G}_n with \mathcal{G}_{n-1} ($\mathcal{G}_{n-1} = \mathcal{G}_{n-1} \cup \mathcal{G}_n$, line 35 of Algorithm 3) to form a new hierarchy $n-1$. In the hierarchy $n-1$, again, we first color a subset of vertices in \mathcal{G}_{n-1} with the same color if: they have the same $|\mathcal{K}_v| = n-1$, the cardinality of such set is $n-1$ and there are no links between any two vertices of such set in the conflict graph. Next, we try to color the uncolored vertices in \mathcal{G}_{n-1} according to the following procedure: 1) randomly pick a vertex v from $\mathcal{W}_1 \subset \mathcal{G}_{n-1}$, 2) color with the same color the chosen vertex v and the other vertices $v' \in \mathcal{W}_2$ whose $|\mathcal{K}_{v'}|$ are “close” to $|\mathcal{K}_v|$ in a greedy manner. Here, \mathcal{W}_1 denotes a set of vertices with “small” $|\mathcal{K}_v|$, $v \in \mathcal{G}_{n-1}$ or “large degree” in $\mathcal{H}_{\mathbf{C}, \mathbf{W}}$ and the value of $a \in [0, 1]$ control the size of \mathcal{W}_1 . Formally, $\mathcal{W}_1 = \{v \in \mathcal{G}_i : \min_{v' \in \mathcal{G}_i} |\mathcal{K}_v| \leq |\mathcal{K}_v| \leq \min_{v' \in \mathcal{G}_i} |\mathcal{K}_v| + \lfloor a (\max_{v' \in \mathcal{G}_i} |\mathcal{K}_v| - \min_{v' \in \mathcal{G}_i} |\mathcal{K}_v|) \rfloor\}$. For example, if $a = 0$, then \mathcal{W}_1 denotes the vertex with the smallest $|\mathcal{K}_v|$. $\mathcal{W}_2 \subset \mathcal{G}_i \setminus \{v\}$ is defined as $\mathcal{W}_2 = \{v' \in \mathcal{Q}_i : \min_{v' \in \mathcal{Q}_i} |\mathcal{K}_{v'}| \leq |\mathcal{K}_{v'}| \leq \min_{v' \in \mathcal{Q}_i} |\mathcal{K}_{v'}| + \lfloor b (\max_{v' \in \mathcal{Q}_i} |\mathcal{K}_{v'}| - \min_{v' \in \mathcal{Q}_i} |\mathcal{K}_{v'}|) \rfloor\}$, where \mathcal{Q}_i is defined in Algorithm 3 and $b \in [0, 1]$. For example, if $b = 0$, then we start from the vertex v' such that $|\mathcal{K}_{v'}| - |\mathcal{K}_v|$ is minimized. Here, we are looking for the independent set with size at least i in the i th hierarchy in a greedy manner. After this second coloring procedure, we union the uncolored vertices with the vertices of the next hierarchy, which in this case, is \mathcal{G}_{n-2} . Then, we repeat the same procedure for all the hierarchies. Finally, in line 39 of Algorithm 3, we use a function called LocalSearch to further reduce the required number of colors. The LocalSearch function is described by Algorithm 4 It can be shown that the complexity of HgLC₁ is given by $O(n|\mathcal{V}|^2)$.

V. SIMULATIONS AND DISCUSSIONS

In this section, we numerically analyze the performance of HgLC for finite file packetization by assuming the random popularity-based (RAP) caching policy in Algorithm 1. We first introduce the achievable rate of the benchmark scheme LFU (Least Frequently Used)⁹, given by:

$$R^{\text{LFU}} = \sum_{f=\min_u \{M_u\}+1}^m \left(1 - \prod_{u \in \mathcal{U}_{\{M_u < f\}}} (1 - q_{f,u})^{L_u} \right), \quad (10)$$

where $\mathcal{U}_{\{M_u < f\}}$ denotes the set of users with $M_u < f$.

For simplicity and to illustrate the effectiveness of HgLC, we consider a homogenous scenario in which users request

⁹LFU discards the least frequently requested file upon the arrival of a new file to a full cache of size M_u files. In the long run, this is equivalent to caching the M_u most popular files.

Algorithm 4 LocalSearch($\mathcal{H}_C, \mathbf{w}, c, \mathcal{C}$)

```

1: for all  $c \in \mathcal{C}$  do
2:   Let  $\mathcal{J}_c$  be the set of vertices whose color is  $c$ ;
3:   Let  $\mathcal{B} = \emptyset$ ;
4:   Let  $\hat{c} = c$ ;
5:   for all  $i \in \mathcal{J}_c$  do
6:      $\mathcal{A} = \emptyset$ ;
7:     for all  $j \in \mathcal{N}(i)$  do
8:        $\mathcal{A} = \mathcal{A} \cup \mathcal{C}[j]$ ;
9:       if  $\mathcal{C} \setminus \mathcal{A} \neq \emptyset$  then
10:         $c'$  is randomly picked from  $\mathcal{C} \setminus \mathcal{A}$ ;
11:         $\hat{c}[i] = c'$ ;
12:         $\mathcal{B} = \mathcal{B} \cup \{i\}$ ;
13:      end if
14:    end for
15:    if  $|\mathcal{B}| = |\mathcal{J}_c|$  then
16:       $c = \hat{c}$ ;
17:       $\mathcal{C} = \mathcal{C} \setminus c$ ;
18:    end if
19:  end for
20: end for
21: return  $c$ ;

```

files according to a common Zipf demand distribution with parameter $\gamma \in \{0.2, 0.4\}$ and all caches have size M files. We assume two types of users. In one case, they represent end devices requesting only one file each ($L = 1$). In a second case, they represent helpers/small-cells, each serving 10 end user devices, and consequently collecting at most $L = 10$ distinct requests. Moreover, we let the caching distribution to be uniform, which means that \mathbf{P} is chosen as a m -dimensional vector taking value of $\frac{1}{m}$.¹⁰

Fig. 3(a) plots the average rate for a network with $n = 80$ users with $\gamma = 0.4$, when $L = 1$ and $B = 200$. Observe how the significant caching gains (with respect to LFU) quantified by the upper bound (GCLC with $B = \infty$) are completely lost when using GCLC with finite packetization. On the other hand, observe how HgLC remarkably preserves most of the promising multiplicative caching gains for the same values of file packetization. For example, in Fig. 3(a), if M doubles from $M = 200$ to $M = 400$, then the rate achieved by HgLC essentially halves from 20 to 10. Furthermore, HgLC is able to achieve a factor of 5 rate reduction from LFU for $M = 500$. For the same regime, it is straightforward to verify that neither GCLC nor LFU exhibit this property.¹¹ Note from Fig. 3(a), that in order to guarantee a rate of 20, GCLC requires a cache size of $M = 500$, while HgLC can reduce the cache size requirement to $M = 200$, a $2.5\times$ cache size reduction. Finally, we notice that the computational time required by HgLC in the scenario of Fig. 3(a), with $B = 200$ and $M = 200$ (20% of the library size), computed as Matlab-cputime /10 on an intel i5 2.4 GHz processor, is around 30s.

¹⁰The caching distribution \mathbf{P}^* can be obtained by minimizing $R^{\text{GCLC}}(\mathbf{P}, \mathbf{Q})$ in (4) among all \mathbf{P} described by a m -dimensional vector taking value in $\{\frac{1}{m}, 0\}$ in practice, as suggested in [5].

¹¹While LFU can only provide an additive caching gain, additive and multiplicative gains may show indistinguishable when M is comparable to the library size m . Hence, one needs to pick a reasonably small M ($\frac{m}{n} < M \ll m$) to observe the multiplicative caching gain of HgLC.

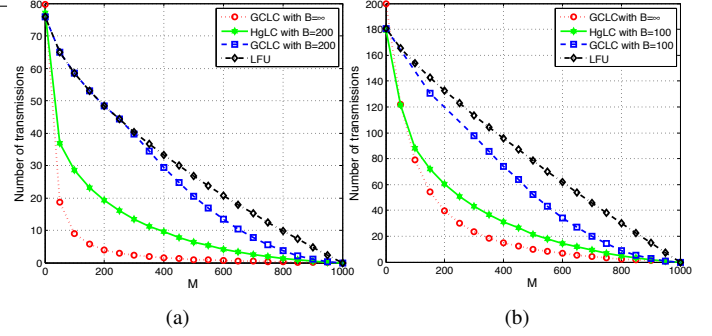


Fig. 3. Average number of transmissions in a heterogeneous shared link network with $m = 1000$. a) $n = 80$, $L = 1$, $\alpha = 0.4$; b) $n = 20$, $L = 10$, $\alpha = 0.2$.

Fig. 3(b), plots the average rate for a network with $n = 20$ helper/small-cell each serving 10 users making requests according to a Zip distribution with $\gamma = 0.2$. Hence the total number of distinct requests per helper is up to $L_u = 10, \forall u \in \{1, \dots, 20\}$. In this case, we assume $B = 100$. Observe first the order-optimal asymptotic rate (shown in red). Note from Figs. 3(a) and 3(b) that when L_u increases (from $L_u = 1$ to $L_u = 10$), while the average rate per request reduces, the gains with respect to LFU also reduce. This is explained by the fact that when aggregating multiple requests per user, there is a higher number of overlapping requests, which increases the opportunities for naive multicasting, as clearly characterized in [2]). Note, however, that HgLC is able to, remarkably, keep similar gains with respect to LFU in this multiple request setting, and approach the asymptotic performance even with just $B = 100$ packets per file, confirming the effectiveness of the local coloring procedures in HgLC.

REFERENCES

- [1] K. Shanmugam, M. Ji, A. Tulino, J. Llorca, and A.G. Dimakis, "Finite Length Analysis of Caching-Aided Coded Multicasting," in *Annual Allerton Conference on Communication, Control, and Computing*, 2014 Proceedings, Oct. 2014.
- [2] M. Ji, A.M. Tulino, J. Llorca, and G. Caire, "Caching and coded multicasting: Multiple groupcast index coding," *arXiv:1402.4572*, 2014.
- [3] J. Llorca, A.M. Tulino, K. Guan, and D. Kilper, "Network-coded caching-aided multicast for efficient content delivery," in *ICC, 2013 Proceedings*. IEEE, 2013.
- [4] M. Ji, G. Caire, and A.F. Molisch, "The throughput-outage tradeoff of wireless one-hop caching networks," *arXiv:1302.2168*, 2013.
- [5] M. Ji, A.M. Tulino, J. Llorca, and G. Caire, "On the average performance of caching and coded multicasting with random demands," in *Wireless Communications Systems (ISWCS), 2014 11th International Symposium on*. IEEE, 2014, pp. 922–926.
- [6] M. Ji, A.M. Tulino, J. Llorca, and G. Caire, "Order-optimal rate of caching and coded multicasting with random demands," *arXiv:1502.03124*, 2013.
- [7] M.A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *Information Theory, IEEE Transactions on*, vol. 60, no. 5, pp. 2856–2867, May 2014.
- [8] M.A. Maddah-Ali and U. Niesen, "Decentralized coded caching attains order-optimal memory-rate tradeoff," *Networking, IEEE/ACM Transactions on*, vol. PP, no. 99, pp. 1–1, 2014.
- [9] M. Ji, G. Caire, and A. F. Molisch, "Wireless device-to-device caching networks: Basic principles and system performance," *arXiv:1305.5216*, 2013.
- [10] Cisco, "The Zettabyte Era-Trends and Analysis," 2013.

- [11] K. Shanmugam, A. G. Dimkakis, and M. Langberg, “Local graph coloring and index coding,” *arXiv:1301.5359*, 2013.